Assessment 1 – Data set

A1844790 - Dylan Rohan

Assignment completed using R and RStudio (R Core Team 2021). Loading libraries (tidyverse, stringr, and dplyr) and uploading/checking csv file data (Wickham 2019; Wickham et al. 2019; Wickham et al. 2021):

library(tidyverse)

library(dplyr)  
library(stringr)

library(rmarkdown)  
  
ashes <- read\_csv("C:\\Users\\rohad\\OneDrive\\Documents\\Data science\\Data Taming, modelling and Vizalization\_RStudio\\a1\\a1\\ashes.csv")

#double slashes for windows directory

ashes <- read\_csv("ashes.csv")

#testing something mentioned in the discussion board, neat!

ashes

#Checking the table, currently a tibble of 27 x 13

unique(ashes$team)

## [1] "England" "English" "Australia"

#need to correct variable English to be England

unique(ashes$role)

## [1] "allrounder" "bowl" "wicketkeeper" "bat" "bowler"   
## [6] "batting" "batsman" "all rounder" "all-rounder"

#many duplicates under alternate variable names, eg. bat, batsman, batting

## Question One: Reading and Cleaning

#### 1.1

For our analysis, the subjects are not the cricketers themselves, but each batting innings they participated in. In order to make the data tidy each subject needs its own row. Rearrange the data into a long format so that there is a row for each batter in each innings. Your new tibble should have 270 rows. [2 points]

Each cell should represent only one measurement. Use str\_match() to create new columns for each of the following for each player innings:

* The batting order, their score, & the number of balls they faced. [2 points]

#1.1  
colnames(ashes)

ashes\_longform <- gather(ashes, key = "innings", value = "description", "Test 1, Innings 1" : "Test 5, Innings 2")  
ashes\_longform

#tibble now in long form, 270 x 5  
ashes\_innings\_first <- ashes\_longform[c(4, 1, 2, 3, 5)]

order <- str\_match(ashes\_innings\_first$description, "Batting at number ..")  
with\_order <- cbind(ashes\_innings\_first, order)  
#Order now has its own column  
runs <- str\_match(with\_order$description, "scored ....")  
with\_runs <- cbind(with\_order, runs)  
#Runs now has its own column  
no.\_of\_balls <- str\_match(with\_runs$description, "from ....")  
all\_columns<- cbind(with\_runs, no.\_of\_balls)  
#no. of balls now has its own column  
batting\_order <- str\_replace\_all(all\_columns$order, "[^0-9.-]", "")  
runs\_ <- str\_replace\_all(all\_columns$runs, "[^0-9.-]", "")  
balls\_ <- str\_replace\_all(all\_columns$no.\_of\_balls, "[^0-9.-]", "")   
#Taking numerical values from strings  
order1 <- tibble(batting\_order)  
runs1 <- tibble(runs\_)  
balls1 <- tibble(balls\_)  
#making data frames from those values  
a1\_o <- cbind(ashes\_innings\_first, order1)  
a1\_o\_r <- cbind(a1\_o, runs1)  
a1\_o\_r\_b<- cbind(a1\_o\_r, balls1)  
#Order same, so binding columns  
a1o\_r\_b <- a1\_o\_r\_b$description <- NULL  
a1\_o\_r\_b <- a1\_o\_r\_b %>%  
 mutate\_all(na\_if, "")  
#removing description column

a1\_o\_r\_b  
#now a tibble of 270 x 7 (removed description, but unused data is still accessible in "ashes\_innings\_first')

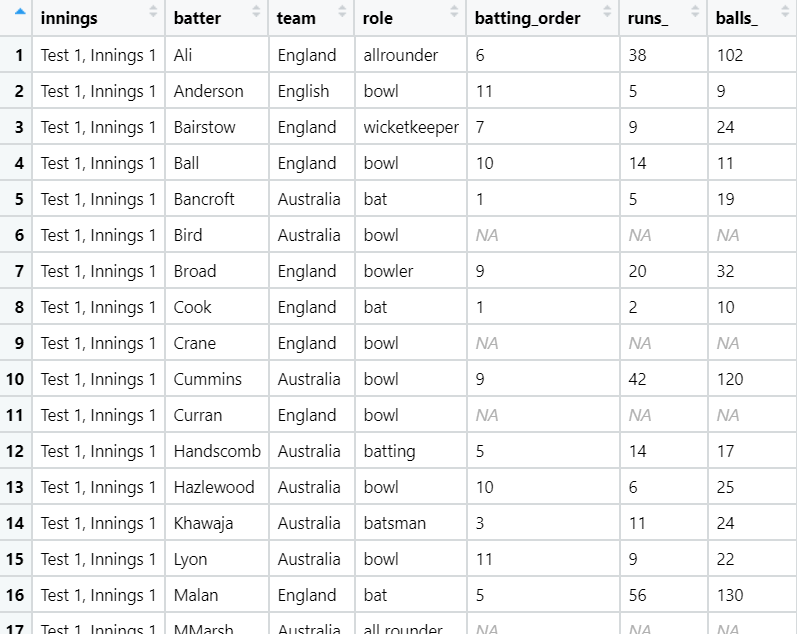


Figure 1: A snippet of the table so far.

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#  
#WAS A KEEN BEEN, SO ALTERNATIVELY:  
trial <- ashes\_innings\_first %>%  
 mutate("runs\_"=str\_match(description,"from ....") , "batting\_order" = str\_match(description, "Batting at number .."), "balls\_" = str\_match(description, "scored ...."))  
#description string broken into appropriate columns  
trial <- trial %>%  
 mutate("runs\_" = str\_replace\_all(trial$runs\_, "[^0-9.-]",""), "balls\_"=str\_replace\_all(trial$balls\_, "[^0-9.-]",""), "batting\_order"=str\_replace\_all(trial$batting\_order, "[^0-9.-]",""))  
trial <- mutate\_all(trial, na\_if, "")  
#Left the description column in here, but all is right with the world  
 # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

#### 1.2

Recode the data to make it ‘tame’, that is:

* Ensure all categorical variables with a small number of levels are coded as factors
  + Innings, team, role, & batting order
* Ensure all categorical variables with a large number of levels are coded as characters,
  + Player
* Ensure all quantitative variables are coded as integers or numeric, as appropriate. [3 points]
  + Runs & balls

ashes\_tibble <- as\_tibble(a1\_o\_r\_b)  
#making a data frame from a1\_o\_r\_b to set value type  
ashes\_tibble$batting\_order <- as.factor(ashes\_tibble$batting\_order)  
#low level ordinal, label = factor  
ashes\_tibble$runs\_ <- as.integer(ashes\_tibble$runs\_)  
ashes\_tibble$balls\_ <- as.integer(ashes\_tibble$balls\_)  
#countable, discrete = integer  
ashes\_tibble$innings <- as.factor(ashes\_tibble$innings)  
#innings total=10, a label/name, low ordered count = factors  
ashes\_tibble <- rename(ashes\_tibble,"player"="batter")  
ashes\_tibble$player <- as.character(ashes\_tibble$player)  
#player makes more sense as a variable name. The teams have several people that could take the position, categorical variable = character  
ashes\_tibble$team <- as.factor(ashes\_tibble$team)  
ashes\_tibble$role <- as.factor(ashes\_tibble$role)  
#both low value labels, so factors

#### 1.3

Clean the data; recode the factors using fct\_recode() such that there are no typographical errors in the team names and player roles. [2 points]

unique(ashes\_tibble$player)

## [1] "Ali" "Anderson" "Bairstow" "Ball" "Bancroft" "Bird"   
## [7] "Broad" "Cook" "Crane" "Cummins" "Curran" "Handscomb"  
## [13] "Hazlewood" "Khawaja" "Lyon" "Malan" "MMarsh" "Overton"   
## [19] "Paine" "Root" "SMarsh" "Smith" "Starc" "Stoneman"   
## [25] "Vince" "Warner" "Woakes"

summary(unique(ashes\_tibble$innings))

## Test 1, Innings 1 Test 1, Innings 2 Test 2, Innings 1 Test 2, Innings 2   
## 1 1 1 1   
## Test 3, Innings 1 Test 3, Innings 2 Test 4, Innings 1 Test 4, Innings 2   
## 1 1 1 1   
## Test 5, Innings 1 Test 5, Innings 2   
## 1 1

summary(unique(ashes\_tibble$team))

## Australia England English   
## 1 1 1

unique(ashes\_tibble$team)

## [1] England English Australia  
## Levels: Australia England English

unique(ashes\_tibble$role)

## [1] allrounder bowl wicketkeeper bat bowler   
## [6] batting batsman all rounder all-rounder   
## 9 Levels: all-rounder all rounder allrounder bat batsman batting ... wicketkeeper

#English to England, unify roles  
ashes\_corrected\_ <- ashes\_tibble %>%  
 mutate(team = fct\_recode(team, "England" = "English"))%>%  
 mutate(role = fct\_recode(role, "all-rounder" = "allrounder", "all-rounder"="all rounder", "batsman"="batting", "batsman"="bat", "bowler"="bowl"))  
ac <- ashes\_corrected\_

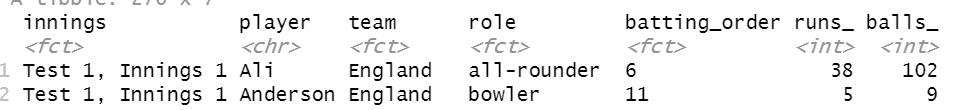


Figure 2: Snippet of the clean and tame tibble.

## Question two: univariate analysis

#### 2.1

Produce a histogram of all scores during the series. [1 point]

#2.1  
#Histogram default below, bin of 30  
ggplot(ac)+geom\_histogram(aes(x=runs\_, ), fill= "black", na.rm=TRUE) +   
 ggtitle("The Runs Achieved Over An Innings in the \n2017/18 Ashes Series")+  
 labs(x= "Scores reached", y ="Frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

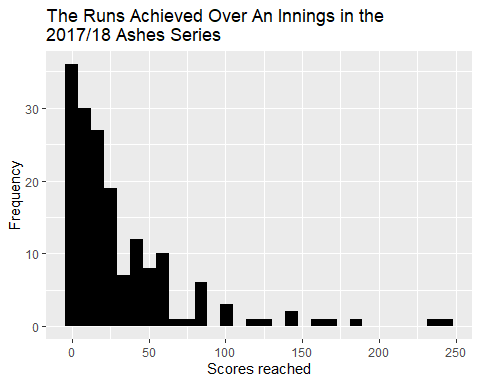


Figure 3: Histogram showing the frequency of scores reached by individual bating sessions across the 2017/18 ashes series.

#ac$runs\_ %>%  
# unique()  
#cool find -> 70 unique values excluding NA, bin of 70 width = 1 for a bar chart as below  
#ggplot(ac)+geom\_histogram(mapping = aes(x=runs\_), na.rm=TRUE, bins=70, binwidth = 1)+  
# ggtitle("Total runs acheieved")+labs(x= "Total runs")

#### 2.2

Describe the distribution of scores, considering shape, location spread and outliers. [4 points]

#2.2  
summary(ac$runs\_, na.rm = TRUE)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 6.00 18.00 32.09 41.00 244.00 101

range(ac$runs\_, na.rm = TRUE, finite= TRUE)

## [1] 0 244

sd(ac$runs\_, na.rm = TRUE)

## [1] 41.30805

table(ac$runs\_)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19   
## 12 4 8 3 9 5 4 3 2 4 3 6 3 2 8 3 1 3 2 2   
## 20 21 22 23 24 25 26 27 28 29 30 31 36 37 38 39 40 41 42 44   
## 5 1 3 1 3 4 3 2 1 2 1 1 4 1 3 2 2 2 2 1   
## 47 49 50 51 53 54 55 56 57 58 61 62 67 76 82 83 86 87 101 102   
## 1 1 1 2 2 1 1 4 1 1 2 1 1 1 1 3 1 1 1 1   
## 103 119 126 140 141 156 171 181 239 244   
## 1 1 1 1 1 1 1 1 1 1

This is a right-skewed shaped graph. The mean score was 32 with a standard deviation of 32. Two players achieved scores over 200 which pulled the mean away from the mode, 12, and median, 18. The interquartile range was 35, the domain was [0,244], and the range was [0,25]. An outlier is anything 1.5 x interquartile range, or IQR, from the edges of the IQR in either direction. Functionally, this indicates that a score higher than (1.5x35+41=) 94 is an outlier. With that definition, there were 11 outliers over the series.

#### 2.3

Produce a bar chart of the teams participating in the series, with different colours for each team. Noting that each player is represented by 10 rows in the data frame, how many players were used by each team in the series? [3 points]

#2.3  
ggplot(ac, aes(x = runs\_, col=team))+geom\_bar()

## Warning: Removed 101 rows containing non-finite values (stat\_count).

#^this maps every players innings, we need to combine player scores across the innings  
indiv\_runs <- ac%>%  
 group\_by(player) %>%  
 summarise(team,role,runs\_in\_series = sum(runs\_, na.rm=TRUE))%>%  
 unique()

## `summarise()` has grouped output by 'player'. You can override using the `.groups` argument.

indiv\_runs

## # A tibble: 27 x 4  
## # Groups: player [27]  
## player team role runs\_in\_series  
## <chr> <fct> <fct> <int>  
## 1 Ali England all-rounder 179  
## 2 Anderson England bowler 8  
## 3 Bairstow England wicketkeeper 306  
## 4 Ball England bowler 15  
## 5 Bancroft Australia batsman 179  
## 6 Bird Australia bowler 4  
## 7 Broad England bowler 136  
## 8 Cook England batsman 376  
## 9 Crane England bowler 6  
## 10 Cummins Australia bowler 166  
## # ... with 17 more rows

unique(ac$player)

## [1] "Ali" "Anderson" "Bairstow" "Ball" "Bancroft" "Bird"   
## [7] "Broad" "Cook" "Crane" "Cummins" "Curran" "Handscomb"  
## [13] "Hazlewood" "Khawaja" "Lyon" "Malan" "MMarsh" "Overton"   
## [19] "Paine" "Root" "SMarsh" "Smith" "Starc" "Stoneman"   
## [25] "Vince" "Warner" "Woakes"

#all players accounted for  
ggplot(indiv\_runs, aes(x=team, fill=team))+  
 geom\_bar()+ggtitle("Number of Players On Each Team in the \n2017/18 Ashes Series")+  
 scale\_y\_continuous(breaks = seq(0, 20, by = 1))+  
 labs(x = "Team", y= "Number of players")

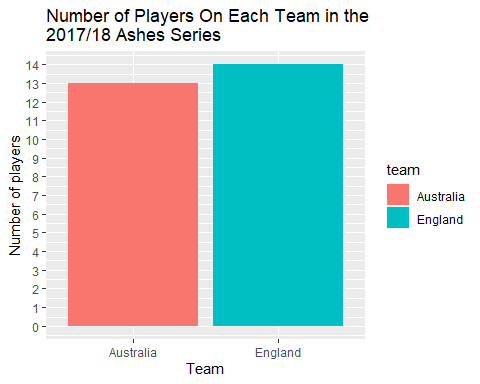


Figure 4: Bar chart indicating the number of players on each team during the 2017/18 Ashes series.

#players per team^   
  
 #\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#  
#What I thought question 2.3 wanted  
indiv\_runs %>%  
 ggplot(aes(x=player, y=runs\_in\_series, fill=role))+  
 geom\_bar(stat="identity")+  
 ggtitle("Individual performance over the \n2017/18 Ashes series")+  
 labs(x = "", y= "")+  
 theme(axis.text.x= element\_text(angle =-90, hjust = 0))

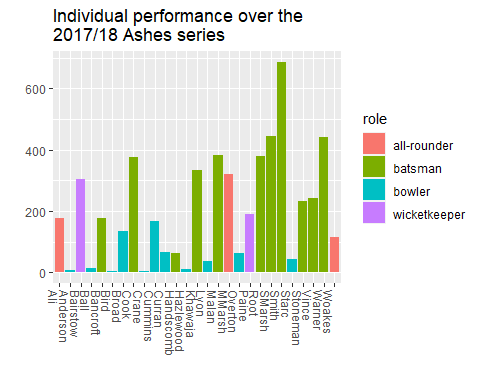


Figure 5: A bonus bar chart of the individual player performances over the 2017/18 Ashes series with colour indicating their role in the team.

#score per player  
 #\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

## Question Three: Scores for each team

#### 3.1

Using ggplot, produce histograms of scores during the series, faceted by team. [1 point]

#3.1  
ac %>%  
 ggplot(aes(x=runs\_, fill=team))+  
 geom\_histogram(show.legend = FALSE)+  
 scale\_y\_continuous(breaks = seq(0, 30, by = 1))+  
 facet\_wrap(~team)+  
 ggtitle("Team Batting Performance in the \n2017/18 Ashes Series")+  
 labs(x = "Score", y= "Frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 101 rows containing non-finite values (stat\_bin).

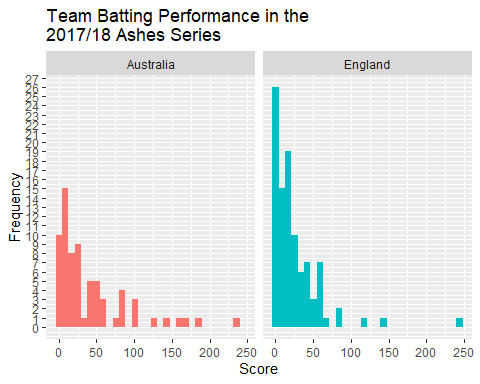


Figure 6: Faceted histograms indicating the frequency of scores reached for each team during the 2017/18 Ashes series; bin=30.

#### 3.2

Produce side-by-side boxplots of scores by each team during the series. [1 point]

**(Side by side as in facet grid? If its just the normal output you’re after see figure 8).**

#3.2  
ac %>%  
 ggplot(aes(y=runs\_, fill=team))+  
 geom\_boxplot(show.legend = FALSE)+  
 facet\_grid(~team)+  
 ggtitle("Boxplot of Team Batting Performance over the \n2017/18 Ashes Series")+  
 labs(x = "Team", y="Runs over the series")

## Warning: Removed 101 rows containing non-finite values (stat\_boxplot).

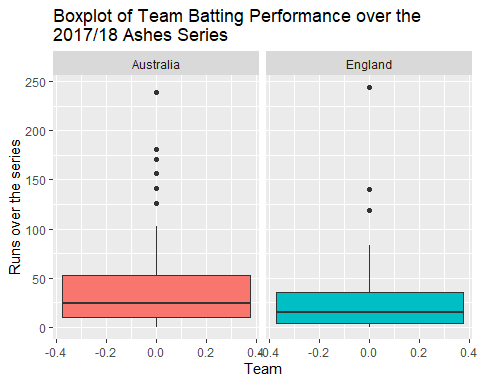


Figure 7: Boxplot representing the runs reached per batsman in an innings over the 2017/18 Ashes series for each team.

#### 3.3

Compare the distributions of scores by each team during the series, considering shape, location, spread and outliers, and referencing the relevant plots. Which team looks to have had a higher average score? [5 points]

#3.3  
#ENGLISH INDIVIDUALS   
england\_players <- ac[ac$team != "Australia", ]  
summary(england\_players$runs\_, na.rm =TRUE)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 4.00 15.00 25.28 36.00 244.00 41

sd(england\_players$runs\_, na.rm = TRUE)

## [1] 33.61336

england\_players %>%  
 arrange(runs\_)

## # A tibble: 140 x 7  
## innings player team role batting\_order runs\_ balls\_  
## <fct> <chr> <fct> <fct> <fct> <int> <int>  
## 1 Test 1, Innings 1 Woakes England all-rounder 8 0 4  
## 2 Test 1, Innings 2 Anderson England bowler 11 0 1  
## 3 Test 2, Innings 1 Anderson England bowler 11 0 3  
## 4 Test 2, Innings 2 Anderson England bowler 11 0 0  
## 5 Test 3, Innings 1 Ali England all-rounder 7 0 2  
## 6 Test 3, Innings 1 Anderson England bowler 11 0 7  
## 7 Test 3, Innings 2 Broad England bowler 10 0 2  
## 8 Test 4, Innings 1 Anderson England bowler 11 0 16  
## 9 Test 5, Innings 1 Anderson England bowler 11 0 3  
## 10 Test 5, Innings 2 Stoneman England batsman 2 0 9  
## # ... with 130 more rows

#England's statistics  
  
#AUSTRALIAN INDIVIDUALS  
aus\_players <- ac[ac$team != "England", ]  
summary(aus\_players$runs\_, na.rm= TRUE)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 10.00 24.00 41.71 52.50 239.00 60

sd(aus\_players$runs\_, na.rm = TRUE)

## [1] 48.88174

aus\_players %>%  
 arrange(runs\_)

## # A tibble: 130 x 7  
## innings player team role batting\_order runs\_ balls\_  
## <fct> <chr> <fct> <fct> <fct> <int> <int>  
## 1 Test 4, Innings 1 Lyon Australia bowler 11 0 10  
## 2 Test 5, Innings 1 Bancroft Australia batsman 1 0 7  
## 3 Test 3, Innings 1 Starc Australia bowler 8 1 3  
## 4 Test 4, Innings 1 Hazlewood Australia bowler 10 1 12  
## 5 Test 2, Innings 2 Hazlewood Australia bowler 11 3 7  
## 6 Test 2, Innings 2 Bancroft Australia batsman 1 4 8  
## 7 Test 3, Innings 1 Lyon Australia bowler 10 4 3  
## 8 Test 4, Innings 1 Bird Australia bowler 9 4 6  
## 9 Test 4, Innings 1 Cummins Australia bowler 8 4 18  
## 10 Test 4, Innings 2 SMarsh Australia batsman 5 4 22  
## # ... with 120 more rows

#for outliers  
ggplot(ac, aes(x = team, y = runs\_, fill =team)) +   
 geom\_boxplot(show.legend = FALSE) +  
 stat\_summary(  
 aes(label = round(stat(y), 1)),  
 geom = "text",   
 fun.y = function(y) { o <- boxplot.stats(y)$out; if(length(o) == 0) NA else o },  
 hjust = -1)+  
 ggtitle("Boxplot of Team Batting Performance over the \n2017/18 Ashes Series")+  
 labs(x = "Team", y="Runs over the series")

## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Removed 101 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 101 rows containing non-finite values (stat\_summary).

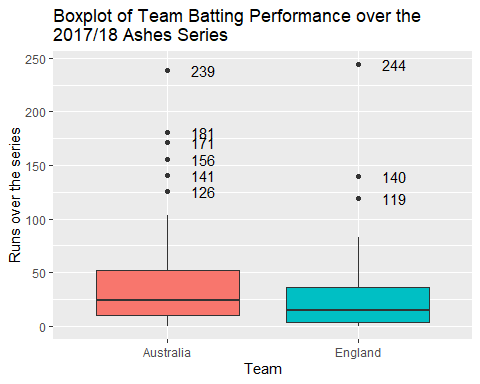


Figure 8: Boxplots representing the spread of runs reached by players in each team over the 2017/18 Ashes seies with outliers labelled with their values.

Both teams have right-skewed histograms, indicating higher scores are less common than lower scores (*figure 6*). England was more right-skewed than Australia due to it having a greater proportion of players reaching lower scores. Using the default bin numbers, the domains are similar for both teams, [0,~250). But ranges differed, England had more players end the innings with scores lower than 50 so their range is much larger, [0, 26]; Australia’s is lower, sitting at [0, 15]. Australia appears to have had the highest average score. The mean score total was located at 42 for Australia with standard deviation of 49 (median of 24), but only 25 for England with a standard deviation of 34 (median 15). The spread also differed, the IQR was 32 for England, and 43 for Australia. This indicates that the English performed more consistently, around a lower mean score while Australia’s scores varied more, but had a few very high scores that pulled the mean higher. According to the boxplots (*figure 7 and 8*), there were six outliers for Australia (126, 141, 156, 171, 181, and 239), and three for England (119, 140, 244). That’s six players that reached a score above (1.5 x IQR + 3rd quartile) 116 for Australia, and three above 84 for England.

## Question Four: Scoring rates

#### 4.1

Produce a scatterplot of scores against number of balls. [1 point]

#4.1  
ggplot(ac, aes( x = runs\_, y= balls\_, col=team))+  
 geom\_point()+  
 geom\_smooth()+  
 ggtitle("Relationship between Balls Faced and \nScore Reached in the 2017/18 Ashes Series")+  
 labs(x = "Score reached", y="Balls")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 101 rows containing non-finite values (stat\_smooth).

## Warning: Removed 101 rows containing missing values (geom\_point).

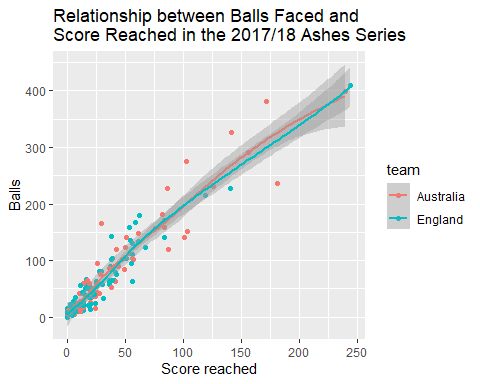


Figure 9: A scatterplot representing the relationship between scores reached and balls received in the 2017/18 Ashes series.

#### 4.2

Describe the relationship between score and number of balls. Are players who face more balls likely to score more runs? [4 points]

There is a positive linear trend for both teams that indicates the more balls faced, the higher the score is likely to be. There are a few things to consider:

1. The first is that you could refuse the first five balls and stay neutral provided you hit a six on the sixth ball. Its therefore quite easy to maintain the ratio of one ball to one run. In this way, there is a lot of room for batters to increase their scores and create a strong positive trendline.
2. A ball can only ever generate a neutral change in score of 0, it cannot reduce the offensive team’s score; so the trendline can only ever be flat, positively trending, or non-existent. Statistically, the offensive team always has the advantage as every ball has the ability to increase the offensive teams score by 0, 1, 4, or 6. The only defense is to get the player out as soon as possible either by bowling them out or catching their hit.
3. This correlation pertains to this specific series. Where skill levels are approximately equivalent and there doesn’t appear to be any contextual factors at first glance that drastically influenced players on game day. But consider that a great bowler could hit the stumps leaving the opposition team with a score of zero, or a defensive batter that could stay in without making a single run, four, or six. This would leave us with a very different correlation. The data from this specific series indicates that more balls will result in more runs, but it’s important to be mindful that assumptions and context that apply here may not be true of other series.

#### 4.3

Compute a new variable, scoring\_rate, defined as the number of runs divided by the number of balls. Produce a scatterplot of scoring\_rate against number of balls. [2 points]

#4.3  
scoring\_rate\_tibble <- ac %>%  
 mutate(scoring\_rates = runs\_/balls\_)  
scoring\_rate\_tibble

## # A tibble: 270 x 8  
## innings player team role batting\_order runs\_ balls\_ scoring\_rates  
## <fct> <chr> <fct> <fct> <fct> <int> <int> <dbl>  
## 1 Test 1, Innings 1 Ali Engl~ all-~ 6 38 102 0.373  
## 2 Test 1, Innings 1 Ander~ Engl~ bowl~ 11 5 9 0.556  
## 3 Test 1, Innings 1 Bairs~ Engl~ wick~ 7 9 24 0.375  
## 4 Test 1, Innings 1 Ball Engl~ bowl~ 10 14 11 1.27   
## 5 Test 1, Innings 1 Bancr~ Aust~ bats~ 1 5 19 0.263  
## 6 Test 1, Innings 1 Bird Aust~ bowl~ <NA> NA NA NA   
## 7 Test 1, Innings 1 Broad Engl~ bowl~ 9 20 32 0.625  
## 8 Test 1, Innings 1 Cook Engl~ bats~ 1 2 10 0.2   
## 9 Test 1, Innings 1 Crane Engl~ bowl~ <NA> NA NA NA   
## 10 Test 1, Innings 1 Cummi~ Aust~ bowl~ 9 42 120 0.35   
## # ... with 260 more rows

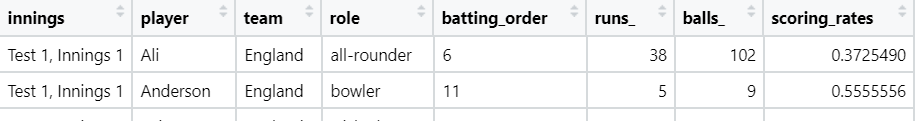


Figure 10: Snippet of the table with the scoring rate column.

#introduced a scoring rate column  
ggplot(scoring\_rate\_tibble, aes( x = scoring\_rates, y= balls\_, col=team))+  
 geom\_point()+  
 geom\_smooth()+  
 ggtitle("Relationship between Balls Faced and \nScoring Rate in the 2017/18 Ashes Series")+  
 labs(x = "Scoring rates (Score/balls faced)", y="Balls faced")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 102 rows containing non-finite values (stat\_smooth).

## Warning: Removed 102 rows containing missing values (geom\_point).

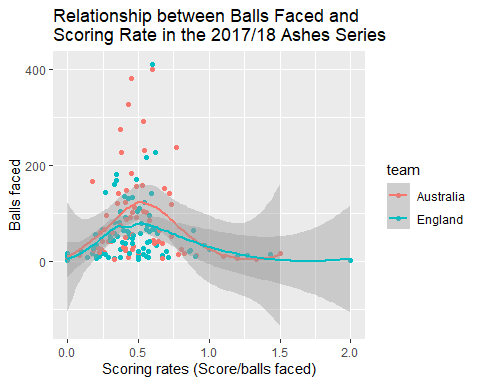


Figure 11: A scatterplot representing the relationship between the scoring rate and the balls received in the 2017/18 Ashes series

#### 4.4

Is there a relationship between scoring rate and number of balls? Are players who face more balls likely to score runs more quickly? [2 points]

Scoring rate and balls faced do not appear to have a linear relationship. Logically, that makes perfect sense. Assuming a large number of balls and approximate skill-level equivalence, perhaps it would make sense for the first few balls to show an improvement in scoring rate as the player warmed up and gets their emotions in check. However, a linear trend would indicate that majority of players somehow improve or, in the case of a negative linear trend, get worse as they play. I wouldn’t expect the best players Australia and England have to offer to do either of those things. Perhaps a new team over hundreds of games, but certainly not by the best of the best in a single test series. Interestingly the geom\_smooth function indicates that there is a negative quadratic relationship with a maximum at an approximate scoring rate of 0.5 at 100 balls. This shape indicates that scoring rates generally increase up until around the 100th ball. Indicating that the sooner the batter is out the better (who’d have thought?). The cause is possibly to do with batting styles. Defensive players let more balls pass by, offensive players take more risks. Perhaps this just indicates the optimal batting style/ risk tolerance for timely ball to score conversion. In any case it’s an interesting point for further investigation.

## Question Five: Teams’ roles

#### 5.1

Produce a bar chart of the number of players on each team participating in the series, with segments coloured by the players’ roles. [1 point]

#5.1  
ggplot(indiv\_runs, aes(x=team, fill=role))+  
 geom\_bar()+  
 ggtitle("Players per Team in the \n2017/18 Ashes Series")+  
 labs(x = "Team", y= "Number of players")

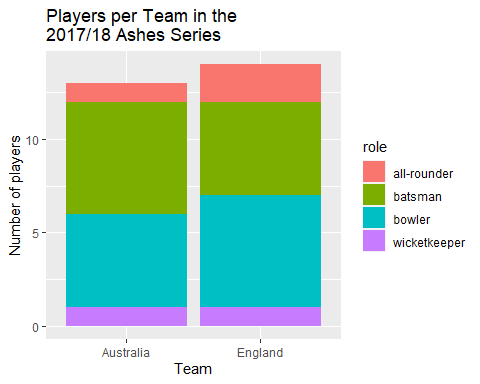


Figure 12: A bar chart representing the number of players on each team with colours indicating the proportion of player roles for that team in the 2017/18 Ashes series.

#### 5.2

Produce a contingency table of the proportion of players from each team who play in each particular role. [2 points]

#5.2  
con\_table <- indiv\_runs%>%  
 group\_by(role) %>%  
 summarise(team, role,player) %>%  
 unique()

## `summarise()` has grouped output by 'role'. You can override using the `.groups` argument.

#keeps 27 subjects and all variables required  
con\_table <- con\_table %>%  
 count(team, role)%>%  
 spread(key = "team", value = n)  
con\_table

## # A tibble: 4 x 3  
## # Groups: role [4]  
## role Australia England  
## <fct> <int> <int>  
## 1 all-rounder 1 2  
## 2 batsman 6 5  
## 3 bowler 5 6  
## 4 wicketkeeper 1 1

#gives a table showing the total players in each roler per team  
ct <- mutate(con\_table, total = sum(Australia+England))  
#adds a column for row totals  
contingency\_table <- ct%>%  
 mutate(Aus=Australia/total, Eng= England/total)  
#adds a column indicating the proportion of each  
contingency\_table <- contingency\_table %>%  
 mutate(Australia = NULL, England =NULL, total=NULL)  
#removes unnecessary columns to reveal the...  
contingency\_table

## # A tibble: 4 x 3  
## # Groups: role [4]  
## role Aus Eng  
## <fct> <dbl> <dbl>  
## 1 all-rounder 0.333 0.667  
## 2 batsman 0.545 0.455  
## 3 bowler 0.455 0.545  
## 4 wicketkeeper 0.5 0.5

#testing something from the tute, found agreement in the outputs

install.packages("gmodels")

library(gmodels)

CrossTable(indiv\_runs$role, indiv\_runs$team)

Table 1: Contingency table indicating the proportion of player roles acorss the two teams.

#### 

#### 5.3

Using these two figures, state which team is made up of a larger proportion of batters, and which team contains a larger proportion of all-rounders. [2 points] [Total: 5 points]

The bar chart shows that Australia opted for an extra batsman, while England opted for an extra bowler. In doing so Australia had more batters. The English also had an additional all-rounder, thus having the highest proportion of them. The contingency table puts numbers to that effect, indicating the proportion of player roles for each team (*table 1*). The proportion of batsman is higher for Australia, and bowler proportions are higher for England *(table 1*). Furthermore, the all-rounder row indicates the English doubled the amount of all rounders held by the Australians, and the number of keepers was equivalent (*table 1)*.

## Question Six: Summary of Insights

Cricket Australia are interested in any insights you can bring with respect to the differences between the two teams, as well as any insights related to scoring. In plain English, write a summary of your key findings from Questions 2-5. Your response should be between 200-250 words. [3 points]

* Scoring rates probably don’t provide a whole lot of meaning without observing how the score was accumulated (which balls were left, which were hit for four or six, which did the players run on). Knowing that will give you an indication of player batting style and perhaps some insight into the optimal batting style. Furthermore, it will provide a frame work for defensive strategy in combatting the different types of playing styles.
* The optimal role proportions can’t accurately be determined from just two teams alone, but with enough data on your team, other teams, and individual playing styles, you might be able to choose team role proportions that have a higher probability of countering the opposing team.
* No consideration has been made as yet to the position of the fielding team. This is perhaps the biggest variable in the game. Where are the defending team, how fast are they, how fast are the batsman, how fast can fielders throw the ball, and what kind of reach are they capable of? These questions could generate a heat map of locations that are likely to result in a batter going out if the ball is hit there. This could offer significant defensive potential and better offensive strategy. It would also enable the team to chose locations that cover the field well, or better yet cover the field in a way that best defends against particular batting styles.

*References:*

R Core Team, 2021, *R: A language and environment for computing and statistical computing* R foundation for statistical computing Vienna, Austria.

Wickham, H 2019, 'stringr: Simple, Consistent Wrappers for Common String Operations', R package version 1.4.0, <<https://CRAN.R-project.org/package=stringr>>.

Wickham, H, Averick, M, Bryan, J, Winston, C, McGowan, LDA, François, R, Grolemund, G, Hayes, A, Henry, L, Hester, J, Kuhn, M, Pedersen, TL, Miller, E, Bache, SM, Müller, K, Ooms, J, Robinson, D, Seidel, DP, Spinu, V, Takahashi, K, Vaughan, D, Wilke, C, Woo, K & Yutani, H 2019, 'Welcome to the {tidyverse}', *Journal of Open Source Software*, vol. 4, no. 43, p. 1686.

Wickham, H, François, R, Henry, L & Müller, K 2021, 'dplyr: A Grammar of Data Manipulation', R package version 1.0.7, <<https://CRAN.R-project.org/package=dplyr>>.

citation()

##   
## To cite R in publications use:  
##   
## R Core Team (2021). R: A language and environment for statistical  
## computing. R Foundation for Statistical Computing, Vienna, Austria.  
## URL https://www.R-project.org/.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {R: A Language and Environment for Statistical Computing},  
## author = {{R Core Team}},  
## organization = {R Foundation for Statistical Computing},  
## address = {Vienna, Austria},  
## year = {2021},  
## url = {https://www.R-project.org/},  
## }  
##   
## We have invested a lot of time and effort in creating R, please cite it  
## when using it for data analysis. See also 'citation("pkgname")' for  
## citing R packages.

citation("tidyverse")

##   
## Wickham et al., (2019). Welcome to the tidyverse. Journal of Open  
## Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Article{,  
## title = {Welcome to the {tidyverse}},  
## author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy D'Agostino McGowan and Romain François and Garrett Grolemund and Alex Hayes and Lionel Henry and Jim Hester and Max Kuhn and Thomas Lin Pedersen and Evan Miller and Stephan Milton Bache and Kirill Müller and Jeroen Ooms and David Robinson and Dana Paige Seidel and Vitalie Spinu and Kohske Takahashi and Davis Vaughan and Claus Wilke and Kara Woo and Hiroaki Yutani},  
## year = {2019},  
## journal = {Journal of Open Source Software},  
## volume = {4},  
## number = {43},  
## pages = {1686},  
## doi = {10.21105/joss.01686},  
## }

citation("dplyr")

##   
## To cite package 'dplyr' in publications use:  
##   
## Hadley Wickham, Romain François, Lionel Henry and Kirill Müller  
## (2021). dplyr: A Grammar of Data Manipulation. R package version  
## 1.0.7. https://CRAN.R-project.org/package=dplyr  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {dplyr: A Grammar of Data Manipulation},  
## author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},  
## year = {2021},  
## note = {R package version 1.0.7},  
## url = {https://CRAN.R-project.org/package=dplyr},  
## }

citation("stringr")

##   
## To cite package 'stringr' in publications use:  
##   
## Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for  
## Common String Operations. R package version 1.4.0.  
## https://CRAN.R-project.org/package=stringr  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {stringr: Simple, Consistent Wrappers for Common String Operations},  
## author = {Hadley Wickham},  
## year = {2019},  
## note = {R package version 1.4.0},  
## url = {https://CRAN.R-project.org/package=stringr},  
## }